CSCI 331:
Introduction to Computer Security

Lecture 22: What I do

Instructor: Dan Barowy

Williams

Announcements

CS Holiday Party

Friday, Dec 8 @ 2:35pm
CS Common Room

Join the CS faculty and your peers for an end-of-semester celebration. We will have hot cocoa and treats for you to enjoy. Last gathering of the year!

Your to-dos

1. Final project, **due Sunday, Dec 10 at 10pm**.
2. Optional book report, now due **Sunday, Dec 17**.
3. Resubmissions due **Sunday, Dec 17**.
4. If you want to talk about your project (or anything else), I have office hours:
   - **Today**, from 4-5:30pm
   - **Friday**, from 12:30-1:30pm

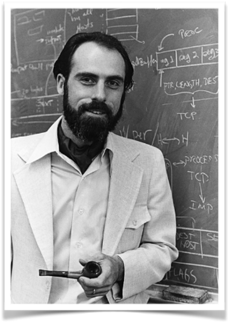
Topics

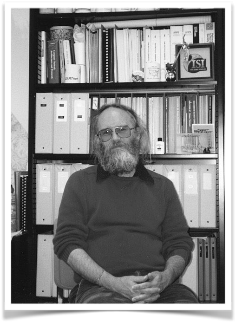
More IP networking

What I do

IP networking, continued

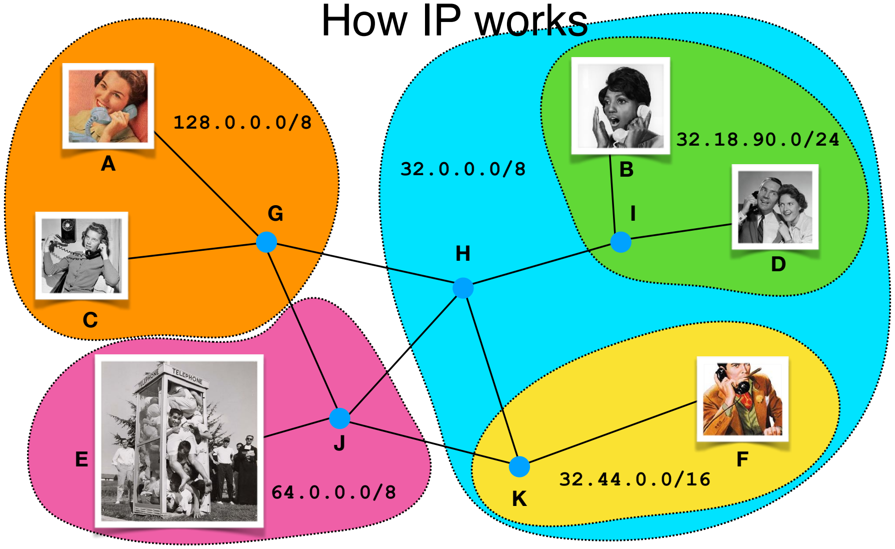
IP networking





- Invented in 1974 by Vint Cerf (Stanford), Bob Kahn (BBN), and Jon Postel (UCLA).
- Key idea: "**connectionless**"
  - instead of connections, do "**packet switching**"

How IP works



- Recall: **A** wants to talk to **F**.

IPv4 address

It's like a **mailing address** for the **Earth**.

32.45.8.12

Each byte ("octet") is between 0 and 255 (0 to 2^8 - 1).

This is actually just a 32-bit number split into 4 pieces.

0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0

32 45 8 12

CIDR

Classless Interdomain Routing ("cider")

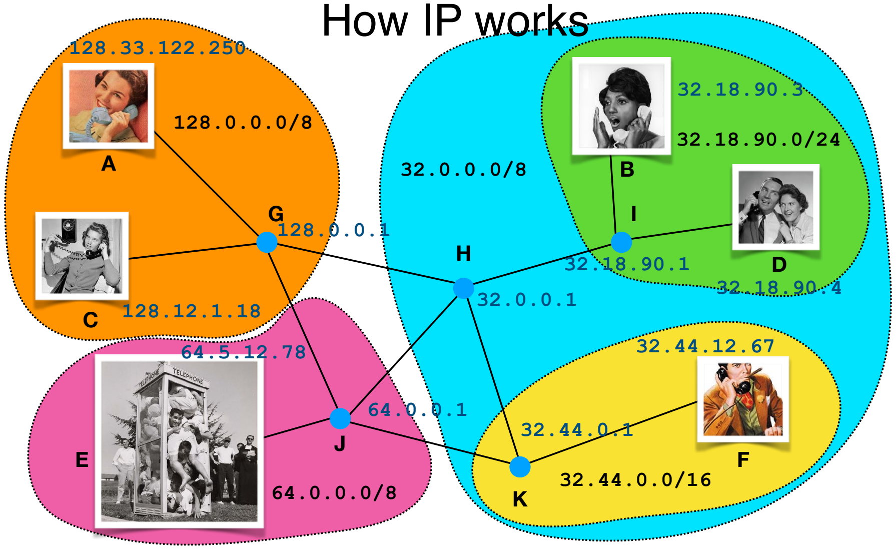
32.0.0.0/8

address prefix subnet mask

addr	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0
mask	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨

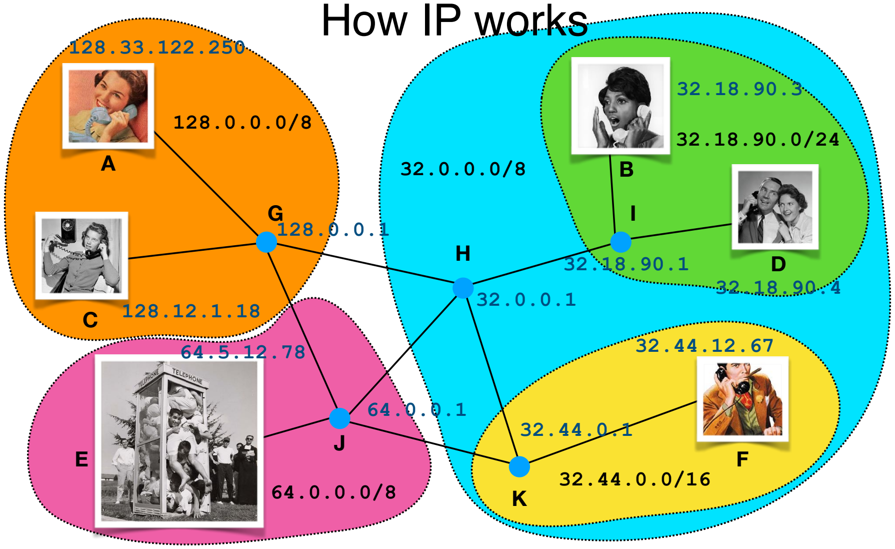
- The subnet mask says which part of the address is fixed, and which part is variable.
- An AS is responsible for routing the variable part.
- In this example, any router knows that the AS for 32.0.0.0/8 is responsible for routing any packet with an address starting with 32.

How IP works



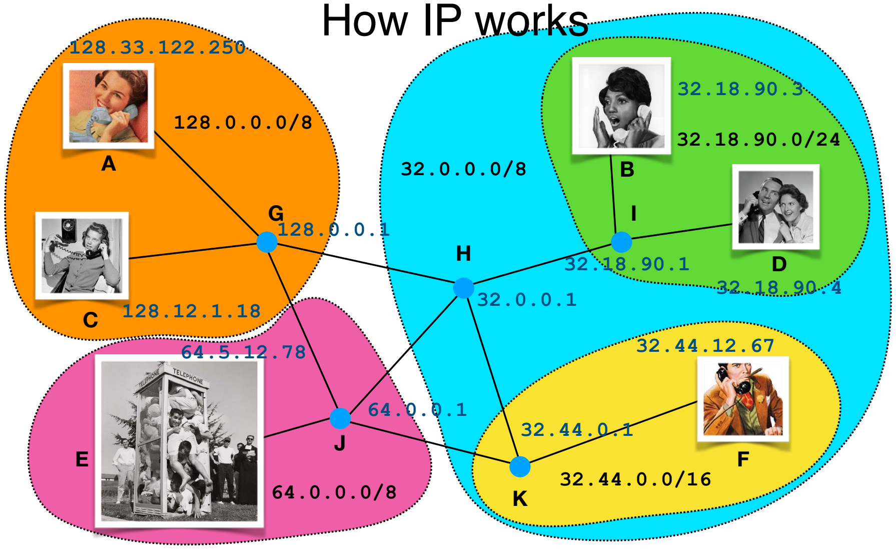
- Every host on the Internet has an IP address.

How IP works



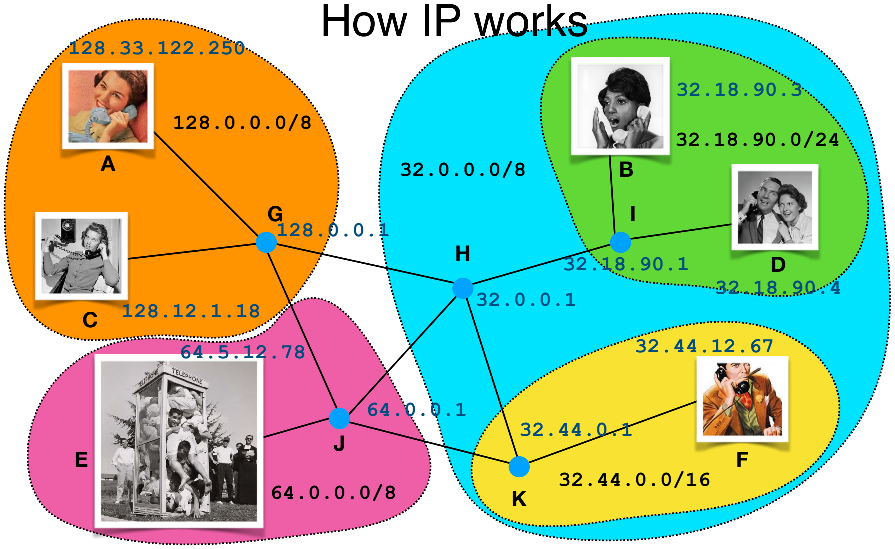
- To route, a router simply needs to find out what routers are responsible for routing packets in a given subnet: this information is stored in a **route table**.

How IP works



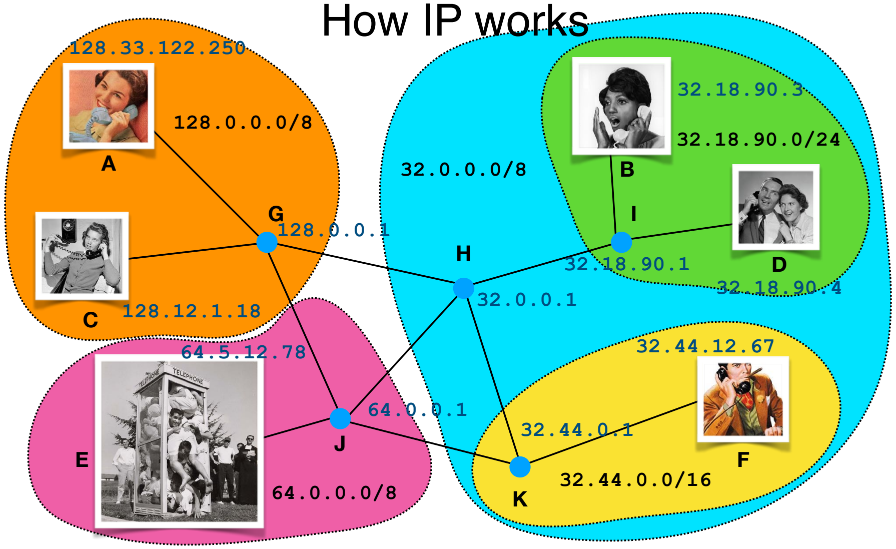
- It then **forwards** the packet to the **next hop** on the way to that subnet.

How IP works



- Tables are only big for a router at the **edge** of a network (e.g., H).

How IP works



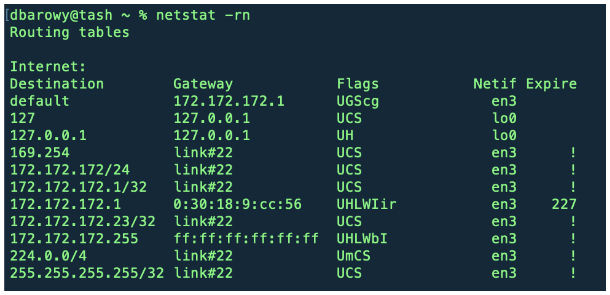
- `128.33.122.250` wants to send to `32.44.12.67`. How?

How IP works

Route table on `128.33.122.250` (A)

Destination	Gateway	Interface	Cost
default	128.0.0.1	en0	1
127.0.0.1	127.0.0.1	lo0	0

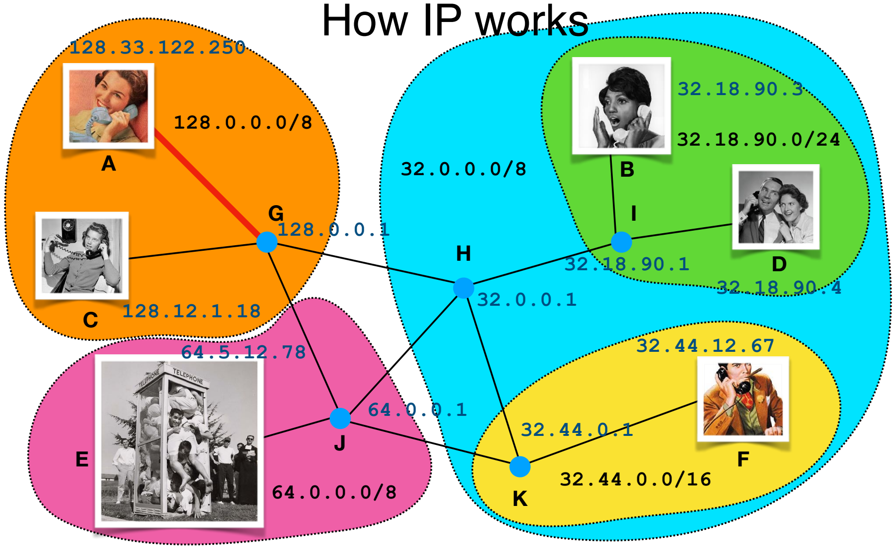
Every device connected to the Internet has a route table.



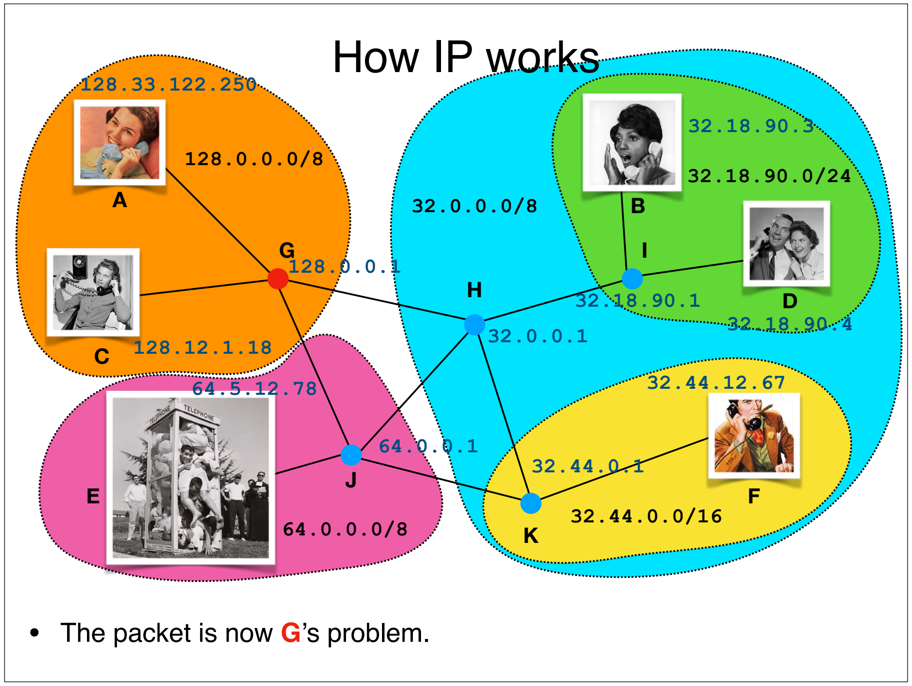
```
[dbarowy@tash ~ % netstat -rn
Routing tables

Internet:
Destination        Gateway            Flags           Netif Expire
default            172.172.172.1      UGScg             en3
127                127.0.0.1          UCS               lo0
127.0.0.1          127.0.0.1          UH                lo0
169.254            link#22            UCS               en3      !
172.172.172/24     link#22            UCS               en3      !
172.172.172.1/32   link#22            UCS               en3      !
172.172.172.1      0:30:18:9:cc:56    UHLWIir           en3    227
172.172.172.23/32  link#22            UCS               en3      !
172.172.172.255    ff:ff:ff:ff:ff:ff  UHLWbI            en3      !
224.0.0/4          link#22            UmCS              en3      !
255.255.255.255/32 link#22            UCS               en3      !
```

How IP works

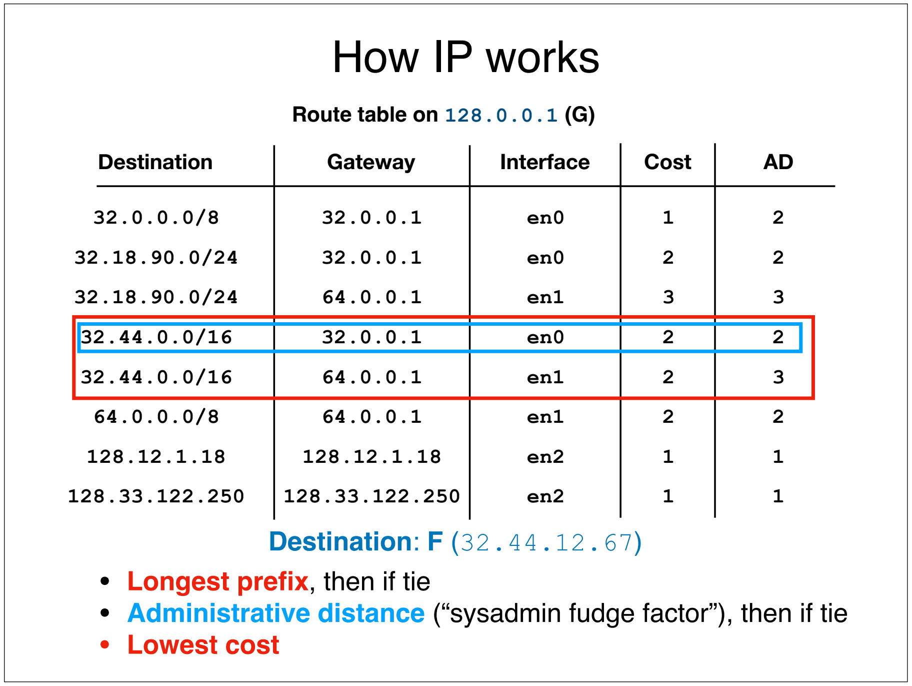


- `128.33.122.250` sends to `128.0.0.1` over `en0`.



How IP works

- The packet is now **G**'s problem.



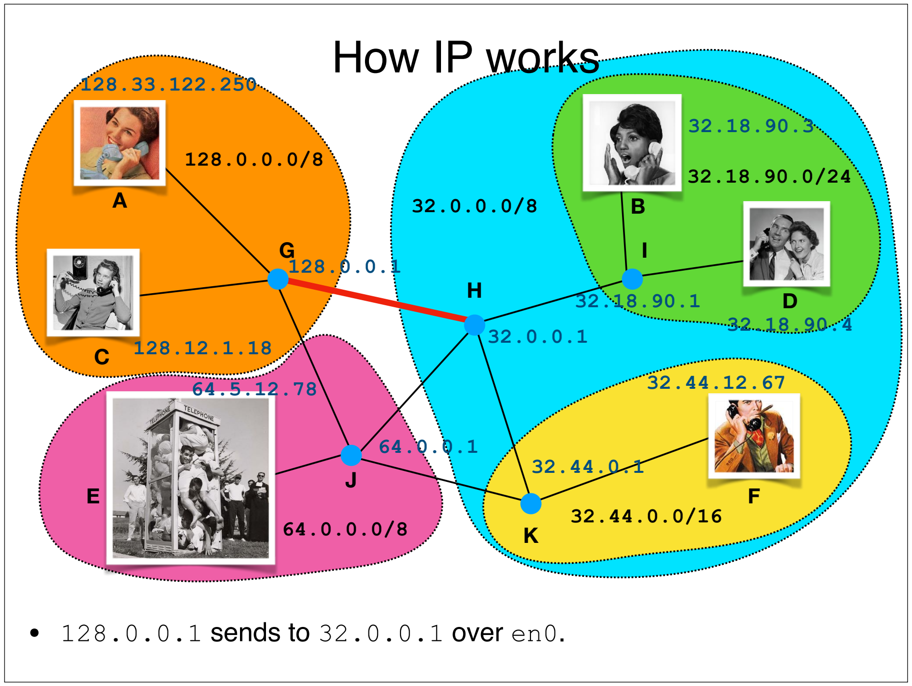
How IP works

Route table on `128.0.0.1` (G)

Destination	Gateway	Interface	Cost	AD
`32.0.0.0/8`	`32.0.0.1`	`en0`	`1`	`2`
`32.18.90.0/24`	`32.0.0.1`	`en0`	`2`	`2`
`32.18.90.0/24`	`64.0.0.1`	`en1`	`3`	`3`
`32.44.0.0/16`	`32.0.0.1`	`en0`	`2`	`2`
`32.44.0.0/16`	`64.0.0.1`	`en1`	`2`	`3`
`64.0.0.0/8`	`64.0.0.1`	`en1`	`2`	`2`
`128.12.1.18`	`128.12.1.18`	`en2`	`1`	`1`
`128.33.122.250`	`128.33.122.250`	`en2`	`1`	`1`

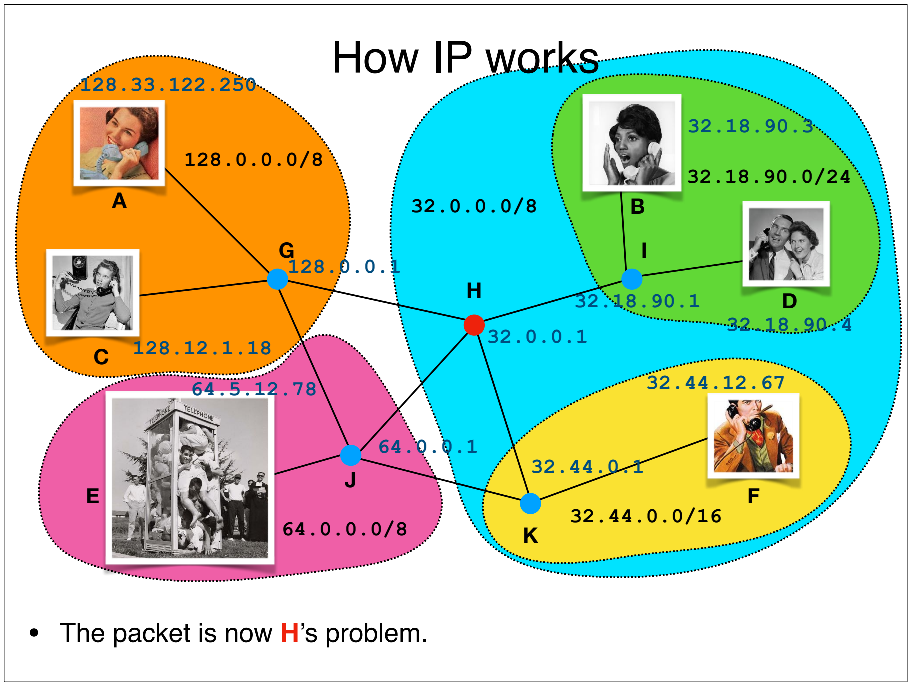
Destination: F (`32.44.12.67`)

- **Longest prefix**, then if tie
- **Administrative distance** ("sysadmin fudge factor"), then if tie
- **Lowest cost**



How IP works

- `128.0.0.1` sends to `32.0.0.1` over `en0`.



How IP works

- The packet is now **H**'s problem.

How IP works

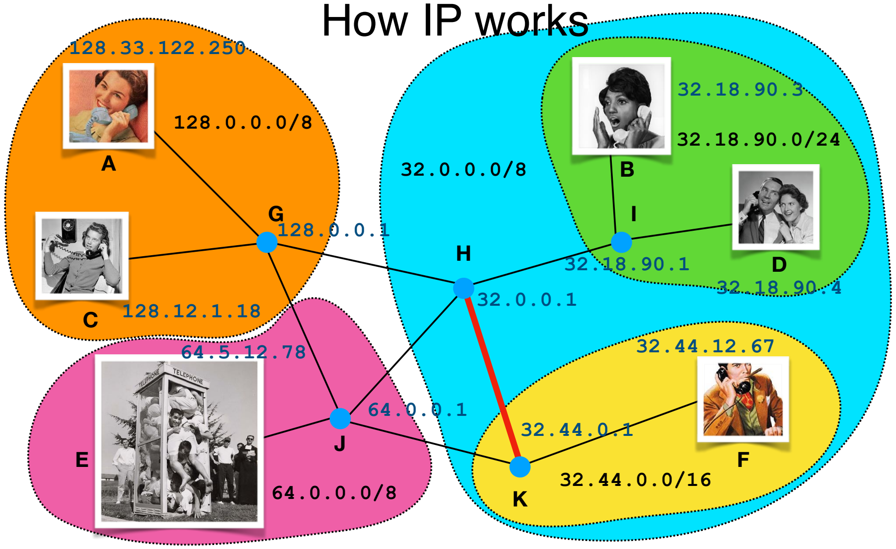
Route table on `32.0.0.1` (H)

Destination	Gateway	Interface	Cost	AD
32.18.90.0/24	32.0.0.1	en4	1	1
32.44.0.0/16	32.0.0.1	en3	1	1
64.0.0.0/8	64.0.0.1	en2	1	2
64.0.0.0/8	128.0.0.1	en1	2	3
128.0.0.0/8	64.0.0.1	en2	2	3
128.0.0.0/8	128.0.0.1	en1	1	2

Destination: F (`32.44.12.67`)

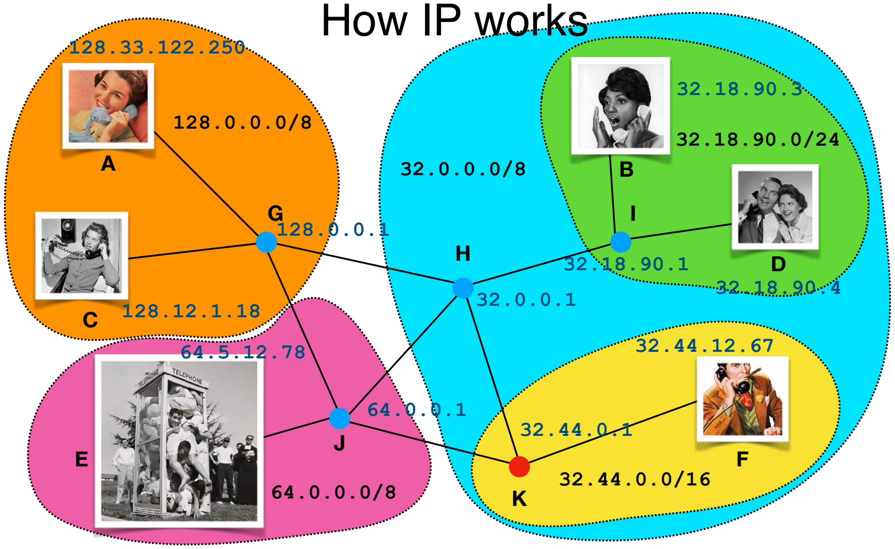
- **Longest prefix**, then if tie
- **Administrative distance** ("sysadmin fudge factor"), then if tie
- **Lowest cost**

How IP works



- `32.0.0.1` sends to `32.44.0.1` over `en3`.

How IP works



- The packet is now **K**'s problem.
- And so on… (**K** is directly connected to **F**)

What is a packet? UDP/IP

<table>
<tr><td></td><td colspan="4">source IP address</td><td rowspan="3">Internet</td></tr>
<tr><td></td><td colspan="4">destination IP address</td></tr>
<tr><td></td><td>pad</td><td>proto</td><td colspan="2">length</td></tr>
<tr><td></td><td colspan="2">source port</td><td colspan="2">dest. port</td><td rowspan="2">transport (UDP)</td></tr>
<tr><td></td><td colspan="2">data length</td><td colspan="2">checksum</td></tr>
<tr><td>32 bits</td><td colspan="4">data</td><td>app (e.g., DNS)</td></tr>
</table>

- **A** wants to send data to **F** of arbitrary length `len`.
- A attaches application info. What source and destination port? (WKP)
- Also length and checksum (to check data transmitted correctly).
- A attaches IP information. What are source and dest. address?
- This (UDP) packet is sent from one host to another.
- Intermediate nodes do not look "inside" the packet. They just forward the data.

What is a packet? UDP/IP

	source IP address			
	destination IP address			Internet
	pad	proto	length	
	source port	dest. port		transport (UDP)
	data length	checksum		
32 bits	data			app (e.g., DNS)

- When a packet arrives at final destination, the Internet part is removed and the rest is handed to the application.
- The application only needs to worry about the transport part.
- UDP only tells the receiver basic information, and if something goes wrong, it's the application's job to handle it.

What is a packet? TCP/IP

	source IP address			
	destination IP address			Internet
	pad	proto	length	
	source port	dest. port		transport (TCP)
	sequence number			
	acknowledgement number			
	flags	length		
	checksum	other		
32 bits	data			app (e.g., HTTP)

- Sometimes we need more reliability.
- TCP is an alternative transport that provides reliability.
- Provides a "pseudo connection" abstraction.
- Ensures that **packets** arrive **in order**, **intact**, with "**best effort**."

How IP works: questions?

Why does networking matter to security?

The Cuckoo’s Egg

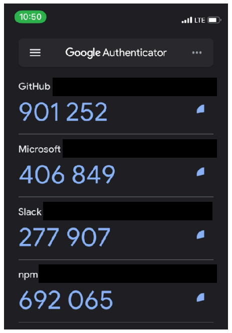
There was a PBS NOVA special on this….

Is hacking easier or harder now?

What I do

Passwords

- Every site: unique email & password
- Password manager or passwords on paper?
- Use 2FA everywhere I can.



GitHub [redacted]
901 252

Microsoft [redacted]
406 849

Slack [redacted]
277 907

npm [redacted]
692 065

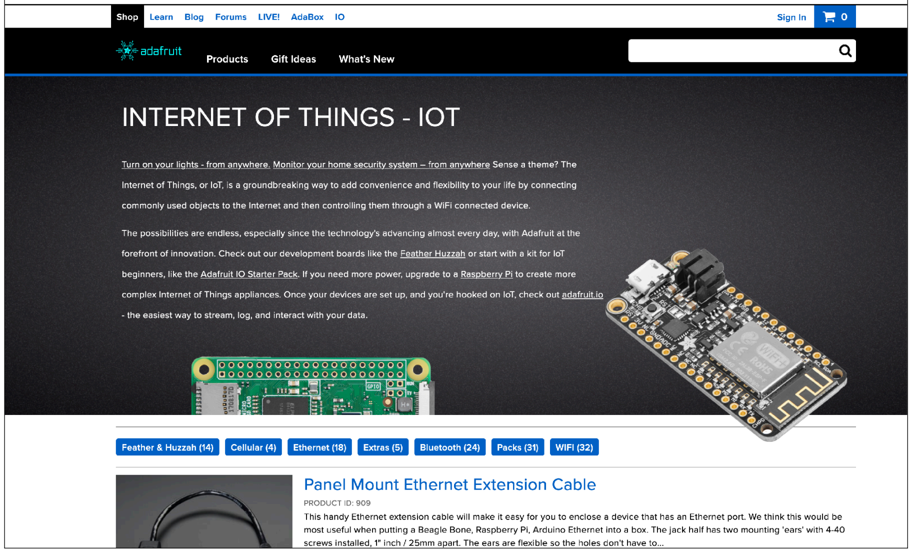
Anti-tracking

- I don’t use Chrome or Edge.
- Ad-blocking extensions:
  - Firefox: uBlock Origin
  - Safari (iOS): Disconnect Premium
- Javascript-blocking extensions:
  - Firefox: NoScript
  - Safari (iOS): sadly… no equivalent
- DNS filtering: Disconnect DNS Privacy
- Home network:
  - OpenBSD firewall running pf
    - Block traditionally bad netblocks
    - Block all traffic from ad networks
    - Many other custom rules (e.g., block IoT devices)

Internet of Things

- Approach this area with great caution.
- Virtually all devices “phone home”.
- Virtually all devices stop getting patches at some point—some of them are never patched!
- Worth noting that IoT devices are just computers attached to a physical device.
  - You all have the skills to roll your own.
  - Many IoT protocols are open standards (i.e., Thread)

Internet of Things



INTERNET OF THINGS - IOT

Turn on your lights - from anywhere. Monitor your home security system – from anywhere Sense a theme? The Internet of Things, or IoT, is a groundbreaking way to add convenience and flexibility to your life by connecting commonly used objects to the Internet and then controlling them through a WiFi connected device.

The possibilities are endless, especially since the technology's advancing almost every day, with Adafruit at the forefront of innovation. Check out our development boards like the Feather Huzzah or start with a kit for IoT beginners, like the Adafruit IO Starter Pack. If you need more power, upgrade to a Raspberry Pi to create more complex Internet of Things appliances. Once your devices are set up, and you're hooked on IoT, check out adafruit.io - the easiest way to stream, log, and interact with your data.

Panel Mount Ethernet Extension Cable

PRODUCT ID: 909

This handy Ethernet extension cable will make it easy for you to enclose a device that has an Ethernet port. We think this would be most useful when putting a Beagle Bone, Raspberry Pi, Arduino Ethernet into a box. The jack half has two mounting 'ears' with 4-40 screws installed, 1" inch / 25mm apart. The ears are flexible so the holes don't have to...

Mobile device security

- Do not show text messages on lock screen
- Generally disable anything on lock screen
- Configure device to wipe itself on failed login attempts
- Only use social media through website, NOT APP.
  - Use Firefox “multi-account containers” to prevent cross-site tracking (e.g., cookies)
- Toss up: “find my device”
  - I use it… because I lose stuff all the time.

Payments

- **Credit cards** are horribly insecure
  - Stealing credit card numbers is trivially easy
  - However
    - Risk is on the bank
    - Maximum loss up to $50
    - Inconvenience of reporting is on you
    - Banks actively monitor fraud.
- **I avoid credit cards.**

Payments

- Do not use **debit cards** except at trusted ATMs
  - Authentication is weak (short PIN)
  - Risk is on the bank, however liability is complicated and maximum loss is much higher (including **ALL YOUR MONEY**).
  - You pay until merchant dispute is resolved.
  - Not as actively monitored as CC.
  - **I avoid them as much as possible.**
  - More info:

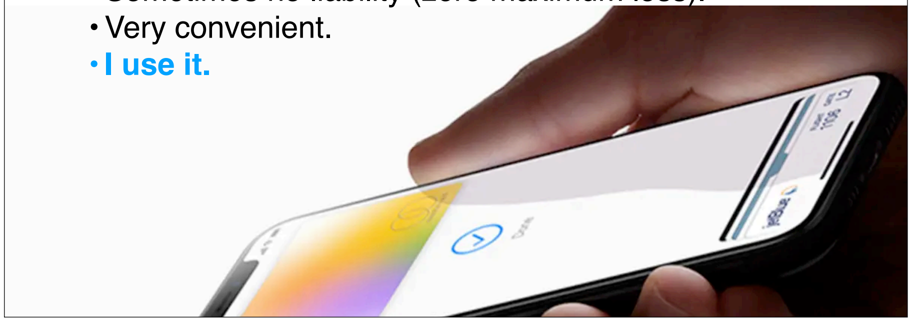
`https://www.consumer.ftc.gov/articles/0213-lost-or-stolen-credit-atm-and-debit-cards`

Payments

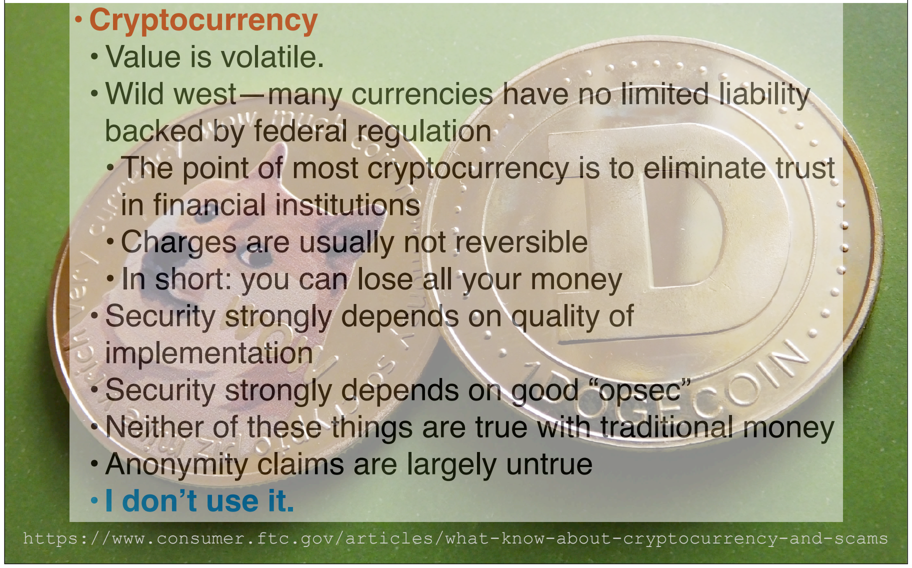
- **Cash** is not terrible
  - Risk is entirely on you, but is manageable (you don’t carry all your money all the time)
  - It is inconvenient.
  - **I sometimes use it.**

Payments

- **NFC** (near-field communication) payments are promising
  - Many different systems— hard to generalize, BUT
  - Risk on bank, but risk is usually lower (hard to steal credentials; varies by vendor).
  - Sometimes no liability (zero maximum loss).
  - Very convenient.
  - **I use it.**



Payments



- **Cryptocurrency**
  - Value is volatile.
  - Wild west—many currencies have no limited liability backed by federal regulation
    - The point of most cryptocurrency is to eliminate trust in financial institutions
    - Charges are usually not reversible
    - In short: you can lose all your money
  - Security strongly depends on quality of implementation
  - Security strongly depends on good “opsec”
  - Neither of these things are true with traditional money
  - Anonymity claims are largely untrue
  - **I don’t use it.**

https://www.consumer.ftc.gov/articles/what-know-about-cryptocurrency-and-scams

Backups

- You should back up your data.
- Make as many copies as you can afford.
- Think short-term and long-term backups.
  - Short-term defends against accidents.
  - Long-term defends against ransomware (and some accidents).
- Long-term backups should be kept off-site.
- Flash memory (SSDs; thumb drives) is not reliable for long-term storage.
  - Use magnetic or optical media.
  - Gold standard: magtape.
  - Next best: BDXL
  - Most practical: hard disks

When I get a new computer

- I turn off all optional services.
- macOS has some.
- Linux has a lot.
- Windows has an insane number.
- Almost none: the BSDs.
  - NetBSD runs on your Raspberry Pi. Try it!

OpenBSD FreeBSD

Unreasonable things I also do

- I don’t like cloud computing—often serves as a “vendor lock-in” mechanism
- I host many of my own services.
  - firewall
  - DIY wifi access points (Raspberry Pi Zeros!)
    - No SSID beacons; passwords required.
  - email
  - filesharing
  - streaming video
  - streaming audio
- I use “weird” architectures.
  - E.g., OpenBSD on ARMv7.

Thanks for the enjoyable semester!

Recap & Next Class

Today we learned:

IP networking

What I do

Next class:

No next class!